

**REMARKS**

Please reconsider the application in view of the following remarks. Applicant thanks the Examiner for carefully considering this application and for indicating that the drawings filed on November 14, 2003 are accepted.

**Disposition of Claims**

Claims 1-38 are pending in the present patent application. Claims 1, 9, 20, and 28 are independent. The remaining claims depend, either directly or indirectly, from claims 1, 9, 20, and 28.

**Rejections under 35 U.S.C. § 102**

Claims 1-38 stand rejected under 35 U.S.C. § 102 as being anticipated by Tamches, “Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels”, University of Wisconsin, 2001 (“Tamches”). To the extent that this rejection may still apply to the pending original claims, the rejection is respectfully traversed.

“A claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987) (emphasis added). Further, “[t]he identical invention must be shown in as complete detail as is contained in the claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989). The Applicant respectfully asserts that Tamches does not expressly or inherently describe each and every element of independent claims 1, 9, 20, and 28.

Independent claim 1 is directed to a method for defining a trace point. More specifically, independent claim 1 recites, in part, defining a trace point representation in a program *source code* and compiling the program *source code* to generate an instrumented program. The Examiner has asserted that Tamches teaches defining trace points in program source code and compiling program source code to generate an instrumented program. *See* Office Action mailed February 9, 2007, p. 2. The Applicant respectfully disagrees. Specifically, the Examiner has asserted that “instrumentation code” is equivalent to program “source code.” *See* Office Action mailed February 9, 2007, p. 2. By asserting that “instrumentation code” is equivalent to program “source code,” the Examiner is completely misconstruing the broadest ordinary meaning of the term program “source code.” Specifically, “source code” refers to “[c]ode written by a programmer in a high-level language and readable by people but not computers.” *See, e.g.,* The American Heritage® Dictionary of the English Language: Fourth Edition, 2000, as cited at <http://www.bartleby.com>. Further, “[s]ource code must be converted to object code or machine language before a computer can read or execute the program.” *See, e.g.,* The American Heritage® Dictionary of the English Language: Fourth Edition, 2000, as cited at <http://www.bartleby.com>. It is clear that source code is readable by people but cannot be read or executed by a program. Thus, in order for a program to execute the source code, the source code must be converted to object code or machine language.

In view the above, “instrumentation code” cannot be construed to be equivalent to “source code,” as “instrumentation code” is program readable and executable based on Tamches’ assertion that instrumentation code can be spliced directly into the kernel. *See* Tamches, pg. 1. Further, Tamches explicitly states that “[a] second important feature of KernInst’s instrumentation mechanisms is that they do not require kernel source code.” *See* Tamches, pg. 50. In view of this, it

is clear that “instrumentation code” as in Tamches is not equivalent to program “source code” as recited in independent claim 1.

Moreover, assuming *arguendo* that instrumentation code is equivalent to program source code, Tamches does not disclose defining a trace point representation in a program source code as recited in independent claim 1. Specifically, claim 1 recites “defining a trace point representation *in a program source code*.” In contrast, Tamches only discloses that instrumentation code is included in a code patch, which is jumped to *from an instrumentation point*. See Tamches, figure 4.1 on pg. 49 and accompanying text. In other words, Tamches only discloses that an instrumentation point is defined *outside the instrumentation code*, which the Examiner asserts is equivalent to the program source code as recited in independent claim 1. However, Tamches does not disclose that the instrumentation point is *in* the instrumentation code. In view of this, it is clear that an instrumentation point outside instrumentation code of Tamches is not equivalent to a trace point defined in program source code as recited in independent claim 1.

Independent claim 9 is directed to a method for enabling a trace point. More specifically, the method comprises, in part, “*obtaining* a tracing function name from a trace object code using a tracing framework.” In contrast, the Examiner admits that Tamches discloses an *instrumentation point* that causes a jump to a code patch. See Office Action mailed February 9, 2007, p. 4. In Tamches, the jump to the code patch facilitates code replacement because instrumentation code can be spliced into the code patch. See Tamches, figure 4.1 on pg. 49 and accompanying text. In short, Tamches describes the *execution* of code during a code splice of instrumentation code into a code patch. However, Tamches does not disclose that a tracing function name, or anything else, is *obtained* from a trace code object. In view of this, it is clear that *executing* code and jumping to

spliced instrumentation code of Tamches is not equivalent to *obtaining* a tracing function name from a trace object code as recited in independent claim 9.

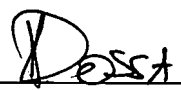
In view of the above, Tamches fails to disclose all the limitations of independent claim 1. In addition, independent claim 20 includes at least the same patentable subject matter as claim 1 and, thus, is patentable as well. Dependent claims 2-8, and 21-27 are allowable for at least the same reasons. Further, Tamches fails to disclose all the limitations of independent claim 9. In addition, independent claim 28 includes at least the same patentable subject matter as claim 9 and, thus, is patentable for the same reasons. Dependent claims 10-19 and 29-38 are allowable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

### Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/350001).

Dated: May 9, 2007

Respectfully submitted,

By  # 20031  
ALY DOSSA  
Robert P. Lord  
Registration No.: 46,479  
OSHA · LIANG LLP  
1221 McKinney St., Suite 2800  
Houston, Texas 77010  
(713) 228-8600  
Attorney for Applicant